UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/764,986	01/22/2001	Andre Chovin	202103US2XPC	4069

22850 7590 04/08/2004

OBLON, SPIVAK, MCCLELLAND, MAIER & NEUSTADT, P.C.
1940 DUKE STREET
ALEXANDRIA, VA 22314

EXAMINER	
BASOM, BLAINE T	

ART UNIT	PAPER NUMBER
2173	8

DATE MAILED: 04/08/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/764,986	CHOVIN ET AL.	
	Examiner	**Art Unit**	
	Blaine Basom	2173	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *28 January 2004*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *2-4 and 8-12* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *2-4 and 8-12* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

BA HUYNH
PRIMARY EXAMINER

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

Response to Arguments

The Examiner acknowledges the Applicants' amendments to claims 2-4, the cancellation of claims 1 and 5-7, and the addition of new claims 8-12 to the present application. The existing claims express, in part, a universal graph compilation system in which is included a spreadsheet associated with a library of codes and corresponding graphical symbols. The Applicants subsequently argue that the combination of Torres (U.S. Patent No. 5,041,131), Mizoguchi (U.S. Patent No. 6,243,858), and Debenedictis (U.S. Patent No. 5,625,823), as presented in the previous Office Action, fails to teach such a feature. Particularly, the Applicants state,

> *...Applicants respectfully point out that Torres fails to disclose that the library further includes codes which correspond with the graphical symbols...Applicants point out that neither Torres, Mizoguchi, or Debenedictis et al. disclose a syntactic and semantic checker which utilizes codes stored in a library to make syntactic and semantic analysis of a generated graph...Applicants point out that compilers may have a syntactic and semantic checker. However, compilers do not inherently make syntactic and semantic checks based on codes stored in a library, wherein graphical symbols are stored in correspondence with the codes used to make the syntactic and semantic check.* (See page 6 of Applicants' Response).

The Examiner respectfully disagrees with these arguments. As is described below, Torres describes a "statistical processing routine" corresponding to each icon in a library of icons. Since the system of Torres is implemented on a personal computer (see column 3, line 60 – column 4, line5), it is understood that this statistical processing routine is necessarily implemented as computer code, so that the microcomputer may process and/or execute the routine. Furthermore, as is shown in the attached appendix, a "routine" is defined by the "Microsoft Computer Dictionary" to incorporate computer code. Thus Torres in fact presents a library of graphical symbols and codes which correspond to the graphical symbols. Assuming, for the sake of argument, that Torres does not teach such codes (for which the Examiner does not

necessarily agree), the Examiner notes that both Mizoguchi and Debenedictis present such codes. As is shown below, Mizoguchi particularly discloses a library of icons, wherein each icon is associated with corresponding code, referred to as a "program module" (see column 1, line 66 – column 2, line 10). Debenedictis presents a system comprising a library of graphical symbols representing elementary component functions, called "tasks," as is described below. Associated with each graphical symbol is corresponding computer code, which is executed in order to implement the elementary component function. As is further shown below, Debenedictis presents a compiler, which is used to compile a graphical representation incorporating the codes, and which as known in the art, inherently comprises a syntactic and semantic checker which ensures that various syntactic and semantic rules are complied with. Since the representation is comprised of the computer codes, these computer codes being compiled by the compiler, it is understood that the syntactic and semantic checker uses these codes to check whether the syntactic and semantic rules of the graph are complied with.

Claim Rejections - 35 USC § 112

The following is a quotation of the first paragraph of 35 U.S.C. 112:

> The specification shall contain a written description of the invention, and of the manner and process of making and using it, in such full, clear, concise, and exact terms as to enable any person skilled in the art to which it pertains, or with which it is most nearly connected, to make and use the same and shall set forth the best mode contemplated by the inventor of carrying out his invention.

Claim10 is rejected under 35 U.S.C. 112, first paragraph, as failing to comply with the enablement requirement. The claim contains subject matter which was not described in the specification in such a way as to enable one skilled in the art to which it pertains, or with which

it is most nearly connected, to make and/or use the invention. Although the specification describes a "code generator" configured to generate computer code, it does not describe or suggest a code generator configured to generate computer code *based on the output of a compiler*, as is expressed in claim 10.

The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

Claim 3 is rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention. Particularly, there is no antecedent basis for "said data and/or commands."

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

Claims 3-4 and 8-12 are rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Patent No. 5,040,131, which is attributed to Torres, over U.S. Patent No. 6,243,858, which is attributed to Mizoguchi et al. (and hereafter referred to as "Mizoguchi"), and also over U.S. Patent No. 5,625,823, which is attributed to Debenedictis et al. (and hereafter referred to as

"Debenedictis"). In general, Torres discusses programs, such as spreadsheets, which allow users to apply various processing functions to data displayed by the programs (see column 1, lines 20-31).

Regarding the claim 8, Torres discloses that a user may apply a function to user-designated data within a spreadsheet by placing an iconic representation of the function into a cell of the spreadsheet (see column 2, lines 14-36). The iconic representation is particularly selected from a "statistical attribute bar," which displays a library of icons representing functions applicable to the spreadsheet data (see column 2, lines 46-60). Upon selection of the icon, a "statistical processing routine" corresponding to the selected icon is loaded and executed in order to determine a statistical attribute, such as the sum or standard deviation, of the designated data (see column 3, lines 26-50; and column 6, line 65 – column 7, line 6). This statistical attribute may then be displayed within a user-designated cell of the spreadsheet by dragging the corresponding icon to the cell (for example, see column 4, line 18 – column 5, line 46). Since this spreadsheet and its functionality may be employed on a microcomputer, specifically a personal computer (see column 3, line 60 – column 4, line5), it is understood that the above-described statistical processing routine is necessarily implemented as computer code, so that the microcomputer may process and/or execute the routine. Torres thus presents a microcomputer, which includes an operating system (see column 3, line 60 – column 4, line 5; and column 7, lines 45-50), and which includes a man-machine interface containing a spreadsheet comprised of a plurality of cells, the spreadsheet associated with a library of graphical symbols and codes which correspond with the graphical symbols, each symbol referring to an elementary function. Torres, however, does not explicitly disclose that this library also comprises symbols

corresponding to links, whereby as expressed in claim 8, the symbols may be selected from the library and placed on the spreadsheet in such a way as to generate a graph, each of the graphical symbols being represented on the spreadsheet in at least one elementary square corresponding to a cell of the spreadsheet, and whereby a connection of the graphical symbol ends at the center of one of the four sides of the cell. Additionally, Torres does not explicitly disclose that the microcomputer is configured to be connected to the memory of at least one component, the component being configured to receive data or commands; that the microcomputer comprises a compiler; that the microcomputer comprises a means for writing to the respective memory of the component, the means for writing being in communication with the compiler via the operating system; and that the microcomputer comprises a syntactic and semantic checker configured to use the codes corresponding to the graphical symbols to check whether the syntactic and semantic rules of the generated graph have been complied with, as is also expressed in claim 8.

Like Torres, Mizoguchi discusses programs which allow users to apply various processing functions to data (see column 1, lines 9-40). Mizoguchi particularly discloses a grid, wherein like the spreadsheet of Torres, the user may place iconic representations of such processing functions into the cells of the grid (see column 1, lines 59-65), and wherein each iconic representation is associated with corresponding code, referred to as a "program module" (see column 1, line 66 – column 2, line 10). Regarding the claimed invention, the user may arrange a plurality of these iconic representations into a graph in order to create a more complex processing function (for example, see column 7, line 25 – column 8, line 26). Each of the icons in the graph are arranged in adjacent cells of the grid, with lines representing connections ending at the centers of the corresponding sides of each cell, as is shown in figure 9A for example. In

addition, Mizoguchi discloses that, in addition to various processing functions, a "conditional branch processing" icon exists (see column 7, lines 39-49), whereby this conditional branch processing icon creates a link between two or more icons, one of which is implemented based on a condition maintained by the conditional branch icon (see column 10, line 56 – column 11, line 30).

Therefore, it would have been obvious to one of ordinary skill in the art, having the teachings of Torres and Mizoguchi before him at the time the invention was made, to modify the spreadsheet taught by Torres such that the icons can be arranged in a graph, which comprises conditional branches, as is taught by Mizoguchi. It would have been advantageous to one of ordinary skill to utilize such a combination because the user would then be able to create *programs* that process the spreadsheet data, thus allowing for larger and more complex processes upon the spreadsheet data, as is demonstrated by Mizoguchi. Thus Torres and Mizoguchi present a microcomputer on which must be implemented the command corresponding to a graph, the microcomputer comprising an operating system, and a man-machine interface containing a spreadsheet comprised of a plurality of cells, wherein the spreadsheet is associated with a library of graphical symbols and codes corresponding to the graphical symbols, the graphical symbols including symbols representing an elementary function and symbols representing links, i.e. conditional branches, relating the elementary function symbols. Such a microcomputer may be employed to generate a graph via selecting graphical symbols from the library and placing them on the spreadsheet, whereby each graphical symbol is represented on the spreadsheet in at least one cell, and whereby the connection line of each graphical symbol in the graph ends at the center of one of the four sides of this cell. The combination of Torres and Mizoguchi, however,

does not explicitly teach configuring this microcomputer to be connected to the memory of at least one component, the component being configured to receive data or commands; that the microcomputer comprises a compiler; that the microcomputer comprises a means for writing to the respective memory of the component, the means for writing being in communication with the compiler via the operating system; and that the microcomputer comprises a syntactic and semantic checker configured to use the codes corresponding to the graphical symbols to check whether the syntactic and semantic rules of the generated graph have been complied with, as is expressed in claim 8.

Like the above-described combination of Torres and Mizoguchi, Debenedictis discusses spreadsheet applications, which allow users to create one or more programs (see column 1, line 63 – column 2, line 13). Debenedictis particularly presents a system, which like the teachings of Mizoguchi, enables the user to create a program by constructing a graph, called an "intertask representation," the graph being comprised of graphical symbols representing elementary component functions, called "tasks" (for example, see figure 19, and its associated description in column 27, lines 1-19). Associated with each graphical symbol is corresponding computer code, which is executed in order to implement the elementary component function (see column 5, line 43 – column 6, line 2). The programs created by the graphs of Debenedictis may be network applications, meaning that the program may be transmitted to one or more network computers, which execute various portions of the program (for example, column 3, line 8 – column 4, line 13; and column 14, lines 42-62). In order to implement such a network application among the various computers of the network, the application is compiled into the various instruction sets of the computers on which the application is implemented (see column 4, lines 14-23). Such

compilers inherently have a syntactic and semantic checker which ensures that various syntactic and semantic rules are complied with, as is known in the art. And since the application is comprised of the computer codes corresponding to the graphical symbols placed within its associated graph (for example, see column 5, line 43 – column 6, line 6), this computer code being compiled by the compiler, it is understood that the syntactic and semantic checker uses these codes corresponding to the symbols to check whether the syntactic and semantic rules of the graph are complied with. Debenedictis thus teaches generating a network application by configuring a computer to be connected to the memory of at least one component, namely a networked computer, the component being configured to receive data or commands; and whereby the computer comprises a compiler and a means for writing to the respective memory of the component, the means for writing being in communication with the compiler via the operating system of the computer, as is known in the art. Debenedictis further teaches that the computer comprises a syntactic and semantic checker configured to use the codes corresponding to the graphical symbols of a graph representing the network application to check whether the syntactic and semantic rules of the generated graph have been complied with.

Therefore, it would have been obvious to one of ordinary skill in the art, having the teachings of Torres, Mizoguchi, and Debenedictis before him at the time the invention was made, to modify the microcomputer and spreadsheet taught by Torres and Mizoguchi, such that the programs created by the spreadsheet may include network applications, like those of Debenedictis. It would have been advantageous to one of ordinary skill to utilize such a combination, because network applications are commonly used in industry, as is demonstrated by Debenedictis (for example, see column 1, lines 16-37). The ability to create such network

programs would thus be beneficial. A microcomputer implementing this spreadsheet of Torres and Mizoguchi, as modified by Debenedictis to implement network applications, is thus configured to be connected to the memory of at least one component, namely a networked computer, the component being configured to receive data or commands; whereby the microcomputer comprises a compiler and a means for writing to the respective memory of the component, the means for writing being in communication with the compiler via the operating system of the microcomputer. Such a compiler comprises a syntactic and semantic checker configured to use codes corresponding to the graphical symbols of a graph representing the network application to check whether the syntactic and semantic rules of the generated graph have been complied with. Consequently, this microcomputer and the network computers with which it is connected, are considered to compose a "universal graph compilation system," like that recited in claim 8.

In reference to claim 3, the spreadsheet of Torres, Mizoguchi, and Debenedictis, is used to create a graph corresponding to a program, which is implemented by one or more remote computers, i.e. components, as is described above. It is understood that these remote computers are connected directly, via a network, to the microcomputer implementing the spreadsheet (for example, see column 3, lines 53-65 of Debenedictis). Consequently, the memories of the network computers are considered to be affixed to the respective network computers and the data and commands of the program are remote loaded.

With respect to claim 4, Mizoguchi discloses that as iconic function representations are dragged onto a grid to create the graph, a check is done to determine if the icon is in an effective place (see column 7, line 66 – column 8, line 22). It is consequently understood that the above-

described spreadsheet of Torres, Mizoguchi, and Debenedictis comprises a "topological checker," which determines if each icon is effectively located in the spreadsheet, as is done by Mizoguchi.

Concerning claims 9-12, the system of Debenedictis generates computer code corresponding to the above-described functional symbols, whereby the code is specifically organized into an intertask representation, and is compiled by a compiler (for example, see column 10, lines 39-67; and column 5, line 43 – column 6, line 2). The microcomputer of Torres, Mizoguchi, and Debenedictis, is therefore considered to comprise a "code generator," like that recited in claim 10. Furthermore, Debenedictis discloses that the code corresponding to each functional symbol may relate to a specific parameter, namely an option, of the elementary component function represented by the symbol (for example, see column 25, lines 10-32; and column 30, lines 3-20). Thus regarding claims 9 and 11, the codes described by Torres, Mizoguchi, and Debenedictis, are understood to relate to a specific parameter of an elementary component function, and are understood to relate to parameterized computer code to be generated by the code generator. Debenedictis further teaches that these codes incorporate a semantic characterization of each connection point of the elementary component function (for example, see column 5, line 43 – column 6, line 46). Consequently, the library codes of Torres, Mizoguchi, and Debenedictis are considered to relate to a semantic characterization of each connection point of the elementary component function in which they represent, as is expressed in claim 12.

Claim 2 is rejected under 35 U.S.C. 103(a) as being unpatentable over the combination of Torres, Mizoguchi, and Debenedictis, which is described above, and also over U.S. Patent No. 6,178,551, which is attributed to Sana et al. (and hereafter referred to as "Sana"). As shown above, Torres, Mizoguchi, and Debenedictis teach a universal graph compilation system, like that of claim 8, whereby a user creates a network program by placing one or more icons representing elementary functions into a spreadsheet. This network program is then automatically and dynamically installed in one or more components, specifically network computers, by transmitting the program from the microcomputer which created the program to the network computers over a network (for example, see column 17, line 25 – column 18, line 27 of Debenedictis). Torres, Mizoguchi, and Debenedictis, however, do not explicitly teach that the network computers have memories which are physically portable and configured to be connected directly to the microcomputer, as is expressed in claim 2.

Like the combination of Torres, Mizoguchi, and Debenedictis, Sana discusses installing programs on network computers via transmitting the programs over a network to the computers (for example, see column 1, lines 32-40). Additionally, and specifically regarding the claimed invention, Sana presents an additional technique used to install programs, whereby particularly, a CD-ROM is used to install the programs (for example, see column 1, lines 20-31). Using such CD-ROMS to install programs is in fact common.

Therefore, it would have been obvious to one of ordinary skill in the art, having the teachings of Torres, Mizoguchi, Debenedictis, and Sana before him at the time the invention was made, to modify the system taught by Torres, Mizoguchi, and Debenedictis, such that the microcomputer may alternatively provide the network program on a CD-ROM, a memory of the

network computer which is physically portable and configured to be connected directly to the microcomputer, whereby this CD-ROM may be used to install the program on the network computer. It would have been advantageous to one of ordinary skill to utilize such a combination because, as taught by Sana, a CD-ROM provides a tangible backup copy of the network program, and provides the user with more control over installation details, such as errors, than does the automatic, dynamic, network installation.

Conclusion

Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Blaine Basom whose telephone number is (703) 305-7694. The examiner can normally be reached on Monday through Friday, from 8:30 am to 5:30 pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, John Cabeca can be reached on (703) 308-3116. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

btb

BA HUYNH
PRIMARY EXAMINER

Microsoft

Appendix

OVER
10,000
ENTRIES

Microsoft Computer Dictionary

Fifth Edition

- Fully updated with the latest technologies, terms, and acronyms
- Easy to read, expertly illustrated
- Definitive coverage of hardware, software, the Internet, and more!

PUBLISHED BY
Microsoft Press
A Division of Microsoft Corporation
One Microsoft Way
Redmond, Washington 98052-6399

Library of Congress Cataloging-in-Publication Data
Microsoft Computer Dictionary.--5th ed.
p. ; cm.
ISBN 0-7356-1495-4
1. Computers--Dictionaries. 2. Microcomputers--Dictionaries.

AQ76.5..M52267 2002
004'.03--dc21 200219714

Printed and bound in the United States of America.

1 2 3 4 5 6 7 8 9 QWT 7 6 5 4 3 2

Distributed in Canada by Penguin Books Canada Limited.

A CIP catalogue record for this book is available from the British Library.

Microsoft Press books are available through booksellers and distributors worldwide. For further information about international editions, contact your local Microsoft Corporation office or contact Microsoft Press International directly at fax (425) 936-7329. Visit our Web site at www.microsoft.com/mspress. Send comments to *mspinput@microsoft.com*.

Acquisitions Editor: Alex Blanton
Project Editor: Sandra Haynes

Body Part No. X08-41929

continuing through referrals to name servers at lower levels of the hierarchy, the DNS is able to match a "friendly" Internet address, such as microsoft.com, with its numerical counterpart, the IP address. Root servers thus contain the data needed for referrals to name servers at the highest level of the hierarchy. There are 13 root servers in the world, located in the United States, the United Kingdom, Sweden, and Japan. *Also called:* root name server. *See also* DNS (definition 1), DNS server, top-level domain.

root web *n.* The default, top-level web provided by a Web server. To access the root web, you supply the URL of the server without specifying a page name or subweb.

ROT13 encryption *n.* A simple encryption method in which each letter is replaced with the letter of the alphabet 13 letters after the original letter, so that A is replaced by N, and so forth; N, in turn, is replaced by A, and Z is replaced by M. ROT13 encryption is not used to protect messages against unauthorized readers; rather, it is used in newsgroups to encode messages that a user may not want to read, such as sexual jokes or spoilers. Some newsreaders can automatically perform ROT13 encryption and decryption at the touch of a key.

rotary dialing *n.* The signaling system used in telephones with rotary dials, in which each digit is associated with a set number of pulses. During dialing, these pulses, which are audible as series of clicks, momentarily turn the current in the telephone wires on and off. *Also called:* pulse dialing. *Compare* touch tone dialing.

rotate *vb.* **1.** To turn a model or other graphical image so that it is viewed at a different angle. **2.** To move bits in a register to the left or to the right. The bit that moves out of the end position rotates to the newly vacated position at the opposite end of the register. *Compare* shift.

rotational delay *n.* The time required for a desired disk sector to rotate to the read/write head. *Also called:* rotational latency.

rotational latency *n.* *See* rotational delay.

RO terminal *n.* Short for read-only **terminal**. A terminal that can receive data but cannot send data. Nearly all printers can be classified as RO terminals.

ROTFL *n.* *See* ROFL.

round *vb.* To shorten the fractional part of a number, increasing the last remaining (rightmost) digit or not, according to whether the deleted portion was over or under five. For example, 0.3333 rounded to two decimal places is 0.33, and 0.6666 is 0.67. Computer programs often round numbers, sometimes causing confusion when the resulting values do not add up "correctly." Percentages in a spreadsheet can thus total 99 percent or 101 percent because of rounding.

round robin *n.* A sequential, cyclical allocation of resources to more than one process or device.

roundtripping *n.* The process of converting files from one format to another for viewing or editing and then converting the files back to the original format again. In some cases, roundtripping can involve repeated conversions of the file from one format to another and back. Frequent roundtripping may be a concern because each conversion has the potential to introduce unwanted changes to the file.

routable protocol *n.* A communications protocol that is used to route data from one network to another by means of a network address and a device address. TCP/IP is an example of a routable protocol.

router *n.* An intermediary device on a communications network that expedites message delivery. On a single network linking many computers through a mesh of possible connections, a router receives transmitted messages and forwards them to their correct destinations over the most efficient available route. On an interconnected set of LANs (local area networks)—including those based on differing architectures and protocols—using the same communications protocols, a router serves the somewhat different function of acting as a link between LANs, enabling messages to be sent from one to another. *See also* bridge, gateway.

routine *n.* Any section of code that can be invoked (executed) within a program. A routine usually has a name (identifier) associated with it and is executed by referencing that name. Related terms (which may or may not be exact synonyms, depending on the context) are *function*, *procedure*, and *subroutine*. *See also* function (definition 3), procedure, subroutine.

routing *n.* The process of forwarding packets between networks from source to destination. *See also* dynamic routing, static routing.

Routing Information Protocol *n.* *See* RIP (definition 1).

routing table *n.* In data communications, a table of information that provides network hardware (bridges and routers) with the directions needed to forward packets of data to locations on other networks. The information contained